

HTML, Javascript und PHP/MYSQL

Inhaltsverzeichnis

1	html	3
2	Javascript	4
2.1	Variablen und Funktionen	4
2.2	Datentypen	4
2.3	Befehle	5
2.4	Kontrollstrukturen	5
2.5	Mathe	5
2.6	DOM	6
3	php ↔ html	7
4	php ↔ javascript	8
5	php ↔ mysql	9
6	php ↔ session	10

1 html

basic

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>new html</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<style type="text/css">.font {...}</style>
<script type="text/javascript" language="JavaScript">
</script>
</head>
<body>content</body>
</html>
```

c.stylesheet

```
<link href="http://yourside.de/style.css" rel="stylesheet" type="text/css">
<style type="text/css">
.font {font-family: Arial, Helvetica, sans-serif;font-size: 12px;}
</style>
```

table

```
<table width="100%" height="333" cellpadding="2" cellspacing="2"
class="tableclass" border="0" bgcolor="#CDCDCD">
<tr><td rowspan="2">A</td><td>B</td><td>C</td></tr>
<tr><td colspan="2">D</td></tr>
<caption class="font">tabelle
```

itemize

```
<li>title<ul>1</ul><ul>2</ul><ul>3</ul></li>
```

link

```
<a href="url" target="_parent">t</a>
<a href="" onMouseOver="proc1()" onClick="proc2()"></a>
```

image+map

```

<map name="map">
<area shape="rect" coords="50,50,30,30" href="#">
<area shape="poly" coords="0,0,30,0,30,30" href="#">
<area shape="circle" coords="50,50,30" href="#">
</map>
```

input

```
<form action="test.html" name="inputaction">
<input type="text" name="text" value="1">
<input type="button" name="button" value="text">
<input type="checkbox" name="checkbox" value="text" checked>
<input type="password" name="password" value="1">
<input type="hidden" name="hiddenfield" value="1">
<input type="radio" name="radio" value="1" checked>
<input type="image" src="misc/html/edit1.gif">
<textarea name="textarea" cols="30" rows="5">text</textarea>
<select name="select">
<option value="1">1</option>
<option value="2">2</option>
</select>
<input type="submit" value="enter">
</form>
```

iframe

```
<iframe src="path/iframe.html" name="iframe" width="100%" height="200"
frameborder="0" scrolling="auto" marginheight="0"></iframe>
<a href="test.html" target="iframe">link</a>
```

divlayer

```
<div id="Layer1" style="position:absolute; width:100px; height:100px; z-index:1; left:0px; top:0px;
visibility:visible; opacity:0.3; border:1px none;"></div>
```

2 Javascript

2.1 Variablen und Funktionen

Variablen sind Objekte mit Name, Datentyp und Wert. In Javascript eine Variable zu definieren, wird der Befehl 'var' benutzt. Bsp: var x=0; Ihr wird keine sichtbarer Datentype zugeordnet, im Vergleich zu Java mit den definierten Datentypen String und int. Variablen können global bzw. lokal verwendet werden, ist aber aus dem Namen nicht erkennbar und leitet sich vom Entstehungsort ab.

Als Funktion werden Objekte mit der Deklaration 'function' bezeichnet. Es werden Werte übergeben, mit mathematischen, oder funktionalen Elementen bearbeitet und möglicherweise mittels 'return' zurückgegeben. In Javascript haben alle Funktionen den gleichen Wert, d.h. eine Hierarchie nach private und public existiert nicht.

2.2 Datentypen

Wie bereits erwähnt kann man den Datentyp nur vom Wert bzw. Index(Array) ableiten. Der einfachste Weg ist, den Prozess bis zur Entstehung der Variable zurück zu verfolgen, um den ursprünglichen Datentyp zu erfahren. Leider kann der ursprüngliche Datentyp verloren gehen. Der Befehl 'typeof' hilft dabei den momentanen

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html><head>
  <title>HTMLDokument</title>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
  <script language="JavaScript" type="text/JavaScript">
    var int=0;
    var string="text";
    var array=[0,1];
    function proc1(x){
      var obj={get func1(){return 0;}}
      setCookie("c1",100);
      var str="<table cellpadding='5'>";
      str=str+"<tr><td width='50'>Integer</td><td width='150'>var int=0;</td><td>"+typeof int+"</td></tr>";
      str=str+"<tr><td>String</td><td>var string='text';</td><td>"+typeof string+"</td></tr>";
      str=str+"<tr><td>Array</td><td>var array=[0,1];</td><td>"+typeof array+"</td></tr>";
      str=str+"<tr><td>Objekt</td><td>var obj={get func1(){return 0;}}</td><td>"+typeof obj.func1+"</td></tr>";
      str=str+"<tr><td>Cookie</td><td>setCookie('c1',0) </td><td>"+typeof getCookie("c1")+"</td></tr></table>";
      document.getElementById("Layer1").innerHTML+="str;
    }
    function setCookie(N,V){document.cookie=N+"="+V+";";}
    function getCookie(N){
      var n=N+"=";
      var C=document.cookie.split(';');
      for(var i=0;i<C.length;i++){
        var c=C[i];
        while(c.charAt(0)==' ')c=c.substring(1);
        if(c.indexOf(n)==0)return c.substring(n.length,c.length);
      }
      return "";
    }
  </script>
</head>
<body>
  <div id="Layer1" style="position:absolute;width:500px;height:100px;left:15px;z-index:1">
    <table cellpadding="5">
      <tr><td width="50">Integer</td><td width="150">var int=0;</td></tr>
      <tr><td>String</td><td>var string="text";</td></tr>
      <tr><td>Array</td><td>var array=[0,1];</td></tr>
      <tr><td>Objekt</td><td>var obj={get func1(){return 0;}}</td></tr>
      <tr><td>Cookie</td><td>setCookie("c1",0) </td></tr>
      <tr><td colspan="2"><a href="" onMouseOver="proc1(100)">berechnen</a></td></tr>
    </table>
  </div>
</body>
</html>
```

2.3 Befehle

Ein Programm ohne Befehle und Anweisungen gibt es nicht und jede noch so grundlegende Aktion benötigt Befehle bzw. Befehlsstrukturen. Ob das Programm ein Packet importiert, eine Variable erstellt, bearbeitet und ausgibt, oder eine Funktion aufgerufen wird, immer sind es Befehle, die die Programmstruktur ermöglichen.

Alle Befehle für Javascript aufzulisten, wäre mit immensen Aufwand verbunden und auch auf einen Einblick in die grundlegenden Befehle, soll an dieser Stelle verzichtet werden.

Im Vordergrund stehen ganz klar die Anweisungen, die sich als Befehle vom 'Document Object Modeling' abheben und nicht als mathematische einzustufen sind.

Befehl	Bedeutung	Beispiel
parseInt	String→Integer	<i>varint1 = parseInt(array[0]);</i>
replace	ersetze mit Wert	<i>d.style.left.replace(/px/,);</i>
split	splitte mit Wert	<i>varstringarray = string.split(" : ");</i>
eval	evaluiere String	<i>eval('varv' + i+' = 0;');</i>
substring	kürze String	<i>string.substring(0, string_couter);</i>

2.4 Kontrollstrukturen

Kontrollstrukturen steuern den Datenfluss anhand von Werten, Parameter und Funktionen. Gleich den Befehlen ist ein Verzicht auf Kontrollstrukturen undenkbar.

```

Werten:    if(x==1){}
Parameter: if(typeof x!='undefined'){}
Funktionen:
function f1(x){return Math.floor(x/2);}
           if(f1(x)>3){}
    
```

Profane Kontrollstrukturen wie grösser, kleiner, gleich, positiv, negativ, etc. werden in der nächsten Tabelle ausser acht gelassen und nur spezielle ausgewählte abgebildet.

Befehl	Bedeutung	Beispiel
typeof	Typ der Variable	<i>if(typeof var != 'undefined'){}</i>
isNaN	is not a Number	<i>if(isNaN(x)){}</i>
indexOf()	ermittelt Index	<i>if(navigator.userAgent.indexOf('Firefox') != -1){}</i>

2.5 Mathe

Es gibt spezielle Befehle die für die Arbeit mit Nummern.

Befehl	Bedeutung	Beispiel
Math.floor()	rundet ab	<i>Math.floor(1/3);</i>
Math.ceil()	rundet auf	<i>Math.ceil(1/3);</i>
Math.random()	Zufallszahl	<i>Math.random() * 10;</i>
Math.abs()	Betrag	<i>Math.abs(-100);</i>

2.6 DOM

Befehl	Bedeutung	Beispiel
<code>getElementById()</code>	Tag über ID	<code>document.getElementById("div1")</code>
<code>getElementsByName()</code>	Tagarray über Tagname	<code>getElementsByName("div");</code>
<code>getElementsByName()[i].id</code>	Tagarray mit Tagname: ID	<code>..ByTagName("div")[i].id</code>
<code>getElementsByName().length</code>	Tagarray: Länge	<code>d.length;</code>
<code>style.left</code>	X Position	<code>d.style.left = 100 + px;</code>
<code>style.top</code>	Y Position	<code>d.style.top = 100 + px;</code>
<code>style.width</code>	Breite	<code>d.style.width = 100 + px;</code>
<code>style.height</code>	Höhe	<code>d.style.height = 100 + px;</code>
<code>style.opacity</code>	Transparenz	<code>d.style.opacity = 1.0;</code>
<code>style.visibility</code>	Sichtbarkeit	<code>d.style.visibility = "visible";</code>
<code>style.zIndex</code>	z-index	<code>d.style.zIndex = 100;</code>
<code>src</code>	Pfad zum Bild	<code>d.src = "img1.jpg";</code>
<code>innerHTML</code>	Taginhalt	<code>d.innerHTML = <div > .. </div > ;</code>
<code>setAttribute</code>	Tagattribut	<code>d.setAttribute("id", "div1");</code>
<code>onmousedown</code>	Event bei Mausklick	<code>document.onmousedown = function(e){}</code>
↔	...	<i>IE : e.clientX FF : e.pageX</i>
<code>ondblclick</code>	Event bei Doppelklick	<code>document.ondblclick = function(e){}</code>
<code>onkeydown</code>	Event bei Tastendruck	<code>document.onkeydown = function(e){}</code>
↔	...	<code>if(e.keyCode == 49){}</code>

3 php ↔ html

```
<?php
echo "
<html>
<head>
<title>html->php</title>
<meta http-equiv=\"Content-Type\" content=\"text/html; charset=iso-8859-1\">
</head>
<body>
<form name="form_name" action="test.php">
<input type="text" name="value1" size="20" value="0">
<input type="hidden" name="value2" value="1">
<input type="submit" value="enter">
</form>
</body>
</html>";
?>
```

```
<?php
$value1=trim(!empty($_POST['value1'])? $_POST['value1']: $_GET['value1']);
$value2=trim(!empty($_POST['value2'])? $_POST['value2']: $_GET['value2']);
echo "
<html>
<head>
<title>get/postHTML in PHP</title>
<meta http-equiv=\"Content-Type\" content=\"text/html; charset=iso-8859-1\">
</head>
<body>
".$value1."
</body>
</html>";
?>
```

4 php ↔ javascript

```
<html>
<head>
  <title>javascript->php</title>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
  <script language="JavaScript" type="text/JavaScript">
    function setSrc(){parent.window.location.href=index.php?value1=0&value2=1;}
  </script>
</head>
<body onLoad="setSrc()">
</body>
</html>

<?php
$value1=0;
echo "
<html>
<head>
  <title>php->javasript</title>
  <meta http-equiv=\"Content-Type\" content=\"text/html; charset=iso-8859-1\">
  <script language=\"JavaScript\" type=\"text/JavaScript\">
    function alertPHPvalue(){alert(\".$value1.\");}
  </script>
</head>
  <body onLoad="alertPHPvalue()">
</body>
</html>";
?>
```


5 php ↔ mysql

```
connection
$con = mysql_connect(host,name,pass);
if(!$con){mysql_error();}

create_database
$result = mysql_query("CREATE DATABASE db_name",$con);
if(!$result){mysql_error();}

show_table
$result = mysql_query("SHOW TABLES FROM name");
if(!$result){mysql_error();}
while ($row = mysql_fetch_row($result)) {echo $row[0];}
mysql_free_result($result);

create_table
$sql = "CREATE TABLE table_name (
user-id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
value1 VARCHAR(30) NOT NULL,
value2 VARCHAR(30) NOT NULL)";
if (!$conn->query($sql)) {echo "Error creating table: " . $conn->error;}

alter_table
$result = mysql_query("ALTER TABLE table_name ADD value3") or die(mysql_error());

insert_table
$sql = "INSERT INTO table_name (value1,value2) VALUES ('text','".str_replace("'",'', $a_value)."')";
if(!($result = $db->sql_query($sql)){message_die(GENERAL_ERROR, 'Could not insert', '', __LINE__, __FILE__, $sql);}

select_table
$sql="SELECT value1,value2 FROM table_name WHERE user-id > 1 AND user-id < 5 ORDER BY value1 DESC LIMIT 1";
if(!($result = $db->sql_query($sql)){mysql_error();}
$selected_value = $db->sql_fetchrow($result);
$value1=$selected_value['value1'];

update_table
$sql = "UPDATE table_name SET value1 = 'a_value', value2 = 'a_value' WHERE user-id > 1 AND user-id < 5";
if(!($result = $db->sql_query($sql)){message_die(GENERAL_ERROR, 'Could not update', '', __LINE__, __FILE__, $sql);}

delete_table
$sql = " DELETE FROM table_name WHERE user-id > 1 AND user-id < 5;
if(!($result = $db->sql_query($sql)){message_die(GENERAL_ERROR, 'Could not delete', '', __LINE__, __FILE__, $sql);}

trop_table
mysql_query('DROP TABLE IF EXISTS '.$name.'.'.$table_name.'') or die(mysql_error());
```

6 php ↔ session

```
<?php
session_start();
$_SESSION['value1']=0;
$php_value=$_SESSION['value1'];
echo $php_value;
session_destroy();
?>
```